

REMARKS

Claims 41 and 43 have been amended. Attached hereto is a marked-up version of the changes made to the claims by the current amendment. The attached page is captioned "Version with markings to show changes made." No claims have been cancelled. No claims have been added. Claims 1-17, 19-41, 43-49, and 51 are pending.

Claims 41 and 43 stand objected to due to typographical errors. Claims 41 and 43 have been amended as suggested in the Office Action. Accordingly, the objection to claims 41 and 43 should be withdrawn.

Claims 1-6, 11-13, 22-27, and 33-35 stand rejected under 35 U.S.C. § 102(a) as being anticipated by CPP (Cambridge Parallel Processing, Gamma II Plus Technical Overview). Claims 32, 40-41, 43-49, and 51 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Fung (U.S. Patent No. 4,380,046). These rejections are respectfully traversed.

The present invention is directed to the connection of a massively parallel processor array to a memory array in a bit serial manner to effect a byte wide data reorganization. Referring to Fig. 2, some computer systems include a main memory 12 which is coupled to both a system CPU 10 via a traditional multi-bit wide bus as well as a massively parallel processing array 14 coupled via a plurality of high speed links to the same main memory 12. The massively parallel processing array 14 typically includes a large plurality of processing elements (PEs) which are arranged as a grid (Fig. 3).

As illustrated in Fig. 4, the plurality of PEs (16a-16n) are typically coupled to the main memory 12 via a corresponding plurality of 1-bit wide data connections 24. Typically, the PEs are designed to read and write data in a vertical direction 30 of the main memory 12. See application at page 4. On the other hand, the CPU 10 of the computer system, accesses the main memory 12 using a traditional multi-bit wide CPU-memory bus and reads and writes the main memory 12 in a horizontal direction 32. See application at page 5. Prior art computer systems therefore must store data in the main memory in

accordance with a data format consistent with one direction (e.g., vertically for efficient access by the array of PEs) and convert the data into another data format consistent with the other direction (e.g., horizontally, for the CPU to transfer between data between main memory and external devices) as necessary depending on what device is accessing the memory. Id. This conversion process may be performed by the PEs, however, the need to convert data format is overhead and reduces the processing throughput of the computer system.

In the present invention the need for the PEs to perform data format conversion is eliminated. Data is stored in the main memory in accordance to one format and if the data must be accessed in another format the conversion is performed “on the fly” by a connection circuit coupled between the PEs and the main memory. See Fig. 5-6. A connection circuit is associated with each PE and includes a plurality of memory buffer registers. The connection circuit can operate in both the horizontal and vertical access modes. In the horizontal access mode, the memory bits are selected so that all bits of a given byte are stored in the same PE (i.e., each set of buffer registers associated with a respective PE contains one byte as seen by the CPU 10 or an external device). In the vertical access mode, each set of buffer registers contains the successive bits at successive locations in the memory corresponding to that PE’s position in the memory word. The selection is achieved utilizing a multiplexer on the input to the register and a pair of tri-state drives which drive each data line.

Accordingly, claims 1 and 22 recite:

a circuit coupled between said main memory and said plurality of processing elements, said circuit writing data from said plurality of processing elements to said memory in a horizontal mode and reading data stored in said main memory in a horizontal mode from said main memory to said plurality of processing elements.

Claim 11 and 33 recite:

a plurality of data path circuits, each of said plurality of data circuits being coupled between said main memory and one of said plurality of processing elements ... wherein each of said data path circuits is adapted to receive data from said respective one of said plurality of processing elements a single bit at a time and write said data

to said main memory in a horizontal mode, and to receive data stored in said main memory in a horizontal mode and output said data to said respective one of said plurality of processing elements a single bit at a time.

Claim 41 recites:

providing a plurality of data bits in a serial manner from said processing element to a data circuit; passing said data through said data circuit; and writing said data to said memory device, wherein said data circuit passes said data directly to said memory device in a horizontal mode

And claim 48 recites:

providing a plurality of data bits from said memory device to a data circuit; passing said data through said data circuit; and outputting said data to said processing element in a serial manner, and wherein at least a portion of said data is stored in said memory device in a vertical mode.

CPP is a technical overview of the architecture, programming languages, and support software of the Gamma II Plus series of computers. CPP, page iii. The Office Action alleges that CPP teaches a circuit coupled between the array memory and the plurality of processing elements which reads data stored in the main memory in a horizontal mode and writes data to the processing elements in a vertical mode. The primary support cited by the Office Action included a section titled "Array Interface" at page 2-13, and a section titled "Data Representation" at page 2-10. The Office Action additionally cited figures 2.1 and 2.6, and a section titled "Master Control Chip" at pages 2-11 through 2-14.

Applicant's representative has reviewed portions of CPP cited above and respectfully asserts that the Office Action has misinterpreted CPP. For example, under "Data Representation" (page 2-10) CPP merely explains the horizontal mode and vertical mode of data storage, and further notes that these modes respectively correspond to the scalar and vector storage modes of a high level language supported by the Gamma II Plus. There is no disclosure or suggestion of circuits recited in independent claims 1, 11, 22, and 33 nor is there any disclosure or suggestion of the method steps recited in independent

claims 41 and 48.

The “Master Control Chip” description at pages 2-11 through 2-14, which includes Figure 2.6, like the “Data Representation” section discussed above, contains language which recognize that the Master Control Chip and the processing elements access memory using different modes, but like the “Data Representation” section, fails to disclose or suggest the claimed circuitry or steps of independent claims 1, 11, 22, 33, 41, and 48. Figure 2.1 is a block diagram of the Gamma II Plus and illustrates the coupling between the Master Control Chip (MCU), and processor elements (PE) as line and therefore also fails to teach or suggest the claimed circuitry and steps recited in independent claims 1, 11, 22, 33, 41, and 48.

Fung is directed to a massively parallel processor computer. Referring to Fig. 1, Fung discloses a computer system including an array 22 of processing elements 44. The array 22 is also coupled to a CPU 29 via bus 29. Column 5, lines 4-10. Significantly, the computer system of Fung does not require conversion between vertical and horizontal modes of data storage. This is because computer system of Fung lacks a main memory. More specifically, the processing element 44 of Fung (shown in greater detail in Fig. 2) includes a processing circuit (comprising counter/shifter 54, logic-slider sub-unit 56, and mask sub-unit 58) which is coupled via a bidirectional single bit bus 52 to a local memory unit 50. The natural data format for each PE of Fung is therefore in the horizontal direction. As such, Fig. 2 shows no data conversion circuit interposed between the local memory unit 50 portion and the processing portion of the processing element. Indeed, since the data access performed by the general purpose CPU 26 is also in the horizontal direction, data is never needed or stored in the vertical direction in the computer system disclosed by Fung. Fung therefore fails to teach or suggest the claimed circuitry and steps recited in independent claims 1, 11, 22, 33, 41 and 48.

Claims 1, 11, 22, 33, 41, and 48 are therefore believed to be allowable over the prior art of record. Claims 2-10 (which depend from claim 1), 12-17 (which depend from claim 11), 23-31 (which depend from claim 22), 34-40 (which depend from claim 33), 43-

Application No.: 09/652,003

Docket No.: M4065.0340/P340

47 (which depend from claim 41), and 49 and 51 (which depend from claim 48) are also believed to be allowable for these reasons and because the combination recited in the claims are not taught or suggested by the prior art of record.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to withdraw the outstanding rejection of the claims and to pass this application to issue.

Dated: April 17, 2003

Respectfully submitted,

By 

Thomas J. D'Amico

Registration No.: 28,371

Christopher S. Chow

Registration No.: 46,493

DICKSTEIN SHAPIRO MORIN &
OSHINSKY LLP

2101 L Street NW

Washington, DC 20037-1526

(202) 785-9700

Attorneys for Applicant

Version With Markings to Show Changes Made

Please amend claims 41 and 43 as follows:

41. A method for writing data from a processing element to a memory device comprising the steps of:

providing a plurality of data bits in a serial manner from said processing element to a data circuit;

passing said data through said data circuit; and

writing said data to said memory device,

wherein said data circuit passes said data directly to said memory device in a horizontal mode and

wherein said step of passing said data further comprises:

outputting each bit of said plurality of data bits from said data circuit on a different data bus associated with said memory device; and

wherein said step of writing said data further comprises writing said each bit of said plurality of [bits] data bits to a location in said memory device associated with a different address.

43. The method according to claim 41, wherein said step of outputting further comprises:

passing each bit of said plurality of data bits through a respective register.